

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)**End of Result Set**

Generate Collection

Print

L3: Entry 1 of 1

File: USPT

Jan 28, 2003

DOCUMENT-IDENTIFIER: US 6513040 B1

TITLE: Method and apparatus for a model data structure for accessing and manipulating java beans

Detailed Description Text (9):

The flexibility of the data structure in the repository model permits several types of association between the ManagedBean Elements and the Managed Beans. For instance, there can be a tree structure organization, in which the Managed Bean Elements of one Managed Bean are distinct from the Managed Bean Elements of another Managed Bean. Or, there can be a directed graph, in which there are distinct Managed Beans, but in which certain Managed Bean Elements belong to more than one Managed Bean. It is also the case that a set of Managed Bean Elements can be common to all Managed Beans in a repository.

Current US Original Classification (1):707/10Current US Cross Reference Classification (1):707/103R

CLAIMS:

1. A method, implemented in a computer system, for creating a storage container for a persistent collection of items, comprising the steps of: storing an object-oriented object in said storage container accessible to a user in said computer system, said storage container having a plurality of related objects in a persistent state; and creating a collection of persistent files as a subclass of said object within said storage container wherein said persistent files represent different types of code and data for said object.
6. An apparatus for creating a storage container for a persistent collection of items, comprising: means for storing an object-oriented object in said storage container accessible to a user in said computer system, said storage container having a plurality of related objects in a persistent state; and means for creating a collection of persistent files as a subclass of said object within said storage container wherein said persistent files represent different types of code and data for said object.
11. A computer program product having a computer readable medium having computer program logic recorded thereon for creating a storage container for a persistent collection of items, comprising: computer readable means for storing an object-oriented object in said storage container accessible to a user in said computer system, said storage container having a plurality of related objects in a persistent state; and computer readable means for creating a collection of persistent files as a subclass of said object within said storage container wherein said persistent files represent different types of code and data for said object.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Hit List

[Clear](#) [Generate Collection](#) [Print](#) [Fwd Refs](#) [Bkwd Refs](#) [Generate OACS](#)

Search Results - Record(s) 1 through 5 of 5 returned.

☐ 1. Document ID: US 20020138353 A1

Using default format because multiple data bases are involved.

L4: Entry 1 of 5

File: PGPB

Sep 26, 2002

PGPUB-DOCUMENT-NUMBER: 20020138353

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20020138353 A1

TITLE: Method and system for analysis of database records having fields with sets

PUBLICATION-DATE: September 26, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Schreiber, Zvi			US	
Gal, Amit			US	

US-CL-CURRENT: 705/26; 707/1

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	-----------	-------

☐ 2. Document ID: US 20020078055 A1

L4: Entry 2 of 5

File: PGPB

Jun 20, 2002

PGPUB-DOCUMENT-NUMBER: 20020078055

PGPUB-FILING-TYPE: new

DOCUMENT-IDENTIFIER: US 20020078055 A1

TITLE: Chaining actions for traversal of a directed graph

PUBLICATION-DATE: June 20, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Bliss, Andrew L.	Redmond	WA	US	
Johns, Kyle R.	Redmond	WA	US	

US-CL-CURRENT: 707/100

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KWIC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	-----------	-------

☐ 3. Document ID: US 6738777 B2

Record List Display

Page 2 of 6

L4: Entry 3 of 5

File: USPT

May 18, 2004

US-PAT-NO: 6738777

DOCUMENT-IDENTIFIER: US 6738777 B2

TITLE: Chaining actions for a directed graph

DATE-ISSUED: May 18, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bliss; Andrew L.	Redmond	WA		
Johns; Kyle R.	Redmond	WA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Microsoft Corporation	Redmond	WA			02

APPL-NO: 09/ 681072 [PALM]

DATE FILED: December 20, 2000

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/100; 707/101

US-CL-CURRENT: 707/100; 707/101

FIELD-OF-SEARCH: 707/100, 707/101

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<u>5524077</u>	June 1996	Faaland et al.	705/8
<u>5751914</u>	May 1998	Coley et al.	706/47
<u>5870545</u>	February 1999	Davis et al.	709/201
<u>5873081</u>	February 1999	Harel	707/3
<u>6292797</u>	September 2001	Tuzhilin et al.	707/6
<u>6324550</u>	November 2001	Huetter	707/206
<u>6341302</u>	January 2002	Celis	709/100

OTHER PUBLICATIONS

Press Release, "TGS Demonstrates HP's DirectModel Technology on SGI, Sun and Windows at SIGGRAPH '97; TGS to Deliver Toolkits and Applications," SIGGRAPH '97, Los Angeles, CA, Aug. 4, 1997, 2 pages.

White Paper: Cripe, Brian E. and Gaskins, Thomas A., The Directmodel Toolkit: Meeting the 3D Graphics Needs of Technical Applications; Date unknown--please admit as prior art. [9 pages].

ART-UNIT: 2171

PRIMARY-EXAMINER: Amsbury; Wayne

ASSISTANT-EXAMINER: Nguyen; CamLinh

ATTY-AGENT-FIRM: Woodcock Washburn LLP

ABSTRACT:

A plurality of actions are applied to a directed graph, where the directed graph has a plurality of nodes. A node in the directed graph is traversed to, and a specific method for each of the plurality of actions is performed on the traversed-node. The traversing and performing are repeated until all nodes in the directed graph have been traversed to. Only a single traversal of the directed graph need be performed to apply all of the actions to the directed graph. The plurality of the actions are in the form of a chain. Each action has an action object pointer, and the actions include a base action at a tail end of the chain and at least one chain action at a head end of the chain. The action object pointer of each chain action points to an immediately adjacent action toward the tail end of the chain, and the action object pointer of the base action points to the chain action at the head end of the chain.

16 Claims, 8 Drawing figures

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	Drawings	Claims	Drawings	Image
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	----------	--------	----------	-------

☐ 4. Document ID: US 6578043 B2

L4: Entry 4 of 5

File: USPT

Jun 10, 2003

US-PAT-NO: 6578043

DOCUMENT-IDENTIFIER: US 6578043 B2

TITLE: Method and apparatus for event modeling

DATE-ISSUED: June 10, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Nye; Jeff	Bellevue	WA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Starwave Corporation	Bellevue	WA			02

APPL-NO: 10/ 022179 [PALM]

DATE FILED: December 12, 2001

PARENT-CASE:

This is a continuation of application Ser. No. 09/170,026, filed Oct. 12, 1998 now U.S. Pat. No. 6,341,279.

INT-CL: [07] G06 F 17/30

US-CL-ISSUED: 707/102; 707/100, 707/101, 707/104

US-CL-CURRENT: 707/102; 707/100, 707/101

FIELD-OF-SEARCH: 707/3, 707/104, 707/10, 707/102, 707/100, 707/101, 717/4, 717/6, 717/122, 717/127, 703/2, 704/224

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
<u>5691917</u>	November 1997	Harrison	717/127
<u>5809235</u>	September 1998	Sharma et al.	709/224
<u>5905890</u>	May 1999	Seaman et al.	717/122
<u>6023572</u>	February 2000	Lautzenheiser et al.	703/2
<u>6065009</u>	May 2000	Leymann et al.	707/10

OTHER PUBLICATIONS

K. Culik II, L.P. Hurd, and S. Yu, Computation Theoretic Aspects of Cellular Automata, Physica D 45, 1990, pp. 357-378.

Stephen Wolfram, Universality and Complexity in Cellular Automata, Physica D 10, 1984, pp. 1-35.

ART-UNIT: 2172

PRIMARY-EXAMINER: Shah; Sanjiv

ATTY-AGENT-FIRM: The Hecker Law Group

ABSTRACT:

The present invention provides a method that allows a developer to add complex dependency logic to an existing database without having to modify the underlying structure of the database. One embodiment of the present invention provides a way to flexibly handle record state transitions by using an event model. The event model is a set of one or more items called an event. Each event in the event model has an associated event type and contains dependency logic that interrelates the events in the event model with one another. Each event represents a set of actions that are optionally contingent upon a condition. The actions and conditions that comprise an event are determine when the event is created. Each event may have a different set of actions and conditions. This enables an event to represent a number of different things. An event can represent anything it is defined to represent. In one embodiment of the present invention an event metamodel is instantiated to represent a number of different event models and the corresponding dependencies that interrelate them. The event metamodel enables the model creator to control what happens to each and every event in the event metamodel without having to modify the underlying structure of the database.

27 Claims, 8 Drawing figures

Full	Title	Citation	Front	Review	Classification	Date	Reference			Claims	KWC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	--	--	--------	-----	-----------	-------

☐ 5. Document ID: US 6553361 B1

L4: Entry 5 of 5

File: USPT

Apr 22, 2003

US-PAT-NO: 6553361

DOCUMENT-IDENTIFIER: US 6553361 B1

TITLE: Knowledge based system

DATE-ISSUED: April 22, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
------	------	-------	----------	---------

Compton; Paul Justin Lloyd	Randwick	AU
Edwards; Glenn Andrew	Nedlands	AU
Lazarus; Leslie	St Ives	AU
Peters; Lindsay Andrew	Eastwood	AU
Harries; Michael Bonnell	Newtown	AU

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP	CODE	COUNTRY	TYPE	CODE
Pacific Knowledge Systems Pty Ltd.	Australian Technology Park				AU		03

APPL-NO: 09/ 618860 [PALM]

DATE FILED: July 18, 2000

PARENT-CASE:

This application claims benefit of provisional appln. Ser. No. 60/144,501 filed Jul. 19, 1999.

INT-CL: [07] G06 F 17/00

US-CL-ISSUED: 706/47; 707/102

US-CL-CURRENT: 706/47; 707/102

FIELD-OF-SEARCH: 706/46, 706/45, 706/47, 707/102

PRIOR-ART-DISCLOSED:

OTHER PUBLICATIONS

Hendra Suryanto et al; The Automatic Compression of Multiple Classification Ripple Down Rule knowledge Based Systems: Preliminary Experiments; Apr. 1999; IEEE; 0-7803-5578; 203-206.*

Compton, P. J. and R. Jansen (1989). "A philosophical basis for knowledge acquisition." 3.sup.rd European Knowledge Acquisition for Knowledge-Based Systems Workshop, Paris: 75-89.

Compton, P., G. Edwards, B. Kang, L. Lazarus, R. Malor, T. Menzies, P. Preston, A. Srinivasan and C. Sammut (1991). Ripple down rules: possibilities and limitations. 6.sup.th Banff AAI Knowledge Acquisition for Knowledge Based Systems Workshop, Banff.

Preton, P., G. Edwards and P. Compton (1994). A 2000 Rule Expert System Without a Knowledge Engineer. Proceedings of the 8.sup.th AAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada.

Preston, P., E. Edwards, P. Compton and D. Litkouhi (1994). An expert System Interpreter for Time Course Data with Refinement in Context, AAI Spring Symposium: Artificial Intelligence in Medicine.

Compton, P., B. Kang, P. Preston and M. Mulholland (1993). Knowledge Acquisition without Analysis. Knowledge Acquisition for Knowledge-Based Systems. Lecture Notes in AI (723(.N. Aussenac, G. Boy, B. Gaines et al. Berlin, Springer Verlag:278-299.

Kang, B., P. Compton and P. Preston (1995). Multiple Classification Ripple Down Rules: Evaluation and Possibilities. Proceedings of the 9.sup.th AAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, University of Calgary.

Compton, P., Z. Ramadan, P. Preston, T. Le-Gia, V. Chellen and M. Mulholland (1998). A trade-off between domain knowledge and problem-solving method power. 11.sup.th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, SRDG Publications, University of Calgary.

Compton, P., P. Preston, G. Edwards and B. Kang (1996). Knowledge based systems that have some idea of their limits. Proceedings of the 10.sup.th AAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada, University of Calgary.

G. Beydoun, A Hoffman, Building Problem Solvers Based on Search Control Knowledge, In Proceedings of Eleventh Workshop on Knowledge Acquisition, Modeling Management, Banff, Canada 1998.

ART-UNIT: 2121

PRIMARY-EXAMINER: Follansbee; John A.

ASSISTANT-EXAMINER: Hirl; Joseph P.

ATTY-AGENT-FIRM: Davis & Bujold, P.L.L.C.

ABSTRACT:

A ripple down rules knowledge based system for generating an output conclusion in response to an input case, the system including a database arranged to receive and store a plurality of conclusions and a plurality of rules, each new conclusion added to the database being associated with a previously stored conclusion as a refinement of the previously stored conclusion and each new conclusion added to the database being associated with a rule, input mechanism for facilitating input of a case, and output mechanism for outputting at least one previously stored conclusion in response to the input case, the system being arranged such that when multiple identical conclusions are returned by the system in response to an input case and a new conclusion associated with the input case is added as a refinement of the identical conclusions, a list of cornerstone conflict cases associated with all of the identical conclusions is generated, and the system being arranged to receive an input rule indicative of a difference between the input case and the cornerstone conflict cases, and to associate the input rule with each of the identical conclusions.

66 Claims, 18 Drawing figures

Full	Title	Citation	Front	Review	Classification	Date	Reference	Claims	KWC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	--------	-----	-----------	-------

Clear

Generate Collection

Print

Fwd Refs

Bkwd Refs

Generate OACS

Term	Documents
PLURALITY	3505707
PLURALITIES	23172
PLURALITYS	6
ACTIONS\$	0
ACTION	2297190
ACTIONA	67
ACTIONAAID	1
ACTIONAAID-NIAL	1
ACTIONAB	3
ACTIONABACILLUS	1
"ACTIONABAUT.THE"	1
(L2 AND (PLURALITY NEAR ACTIONS\$)).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	5

There are more results than shown above. [Click here to view the entire set.](#)



Display Format:

Change Format

[Previous Page](#)[Next Page](#)[Go to Doc#](#)

Freeform Search

Database:	US Pre-Grant Publication Full-Text Database
	US Patents Full-Text Database
	US OCR Full-Text Database
	EPO Abstracts Database
	JPO Abstracts Database
	Derwent World Patents Index
	IBM Technical Disclosure Bulletins

Term:	L1 and container\$ and (directed near graph)	
		

Display:	<input type="text" value="50"/>	Documents in Display Format:	<input type="text" value="FRO"/>	Starting with Number	<input type="text" value="1"/>
-----------------	---------------------------------	-------------------------------------	----------------------------------	-----------------------------	--------------------------------

Generate: ☐ Hit List ☒ Hit Count ☐ Side by Side ☐ Image

Search

Clear

Interrupt

Search History

DATE: Friday, December 10, 2004 [Printable Copy](#) [Create Case](#)

Set Name Query

side by side

Hit Count Set Name

result set



DB=USPT; PLUR=YES; OP=OR

<u>L11</u>	6341302.pn.	1	<u>L11</u>
<u>L10</u>	6324550.pn.	1	<u>L10</u>
<u>L9</u>	6292797.pn.	1	<u>L9</u>
<u>L8</u>	5873081.pn.	1	<u>L8</u>
<u>L7</u>	5870545.pn.	1	<u>L7</u>
<u>L6</u>	5751914.pn.	1	<u>L6</u>
<u>L5</u>	5524077.pn.	1	<u>L5</u>
<u>L4</u>	L1 and container\$ and (directed near graph)	18	<u>L4</u>
<u>L3</u>	L2 and (directed near graph)	1	<u>L3</u>
<u>L2</u>	L1 and (plurality near container\$)	35	<u>L2</u>
<u>L1</u>	707/\$.ccls.	14009	<u>L1</u>

END OF SEARCH HISTORY

Freeform Search

Database:	US Pre-Grant Publication Full-Text Database
	US Patents Full-Text Database
	US OCR Full-Text Database
	EPO Abstracts Database
	JPO Abstracts Database
	Derwent World Patents Index
	IBM Technical Disclosure Bulletins

Term:	L1 and container\$ and (directed near graph)	
		

Display:	<input type="text" value="50"/>	Documents in Display Format:	<input type="text" value="FRO"/>	Starting with Number	<input type="text" value="1"/>
-----------------	---------------------------------	-------------------------------------	----------------------------------	-----------------------------	--------------------------------

Generate: ☐ Hit List ☒ Hit Count ☐ Side by Side ☐ Image

Search

Clear

Interrupt

Search History

DATE: Friday, December 10, 2004 [Printable Copy](#) [Create Case](#)

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=OR

<u>L11</u>	6341302.pn.	1	<u>L11</u>
<u>L10</u>	6324550.pn.	1	<u>L10</u>
<u>L9</u>	6292797.pn.	1	<u>L9</u>
<u>L8</u>	5873081.pn.	1	<u>L8</u>
<u>L7</u>	5870545.pn.	1	<u>L7</u>
<u>L6</u>	5751914.pn.	1	<u>L6</u>
<u>L5</u>	5524077.pn.	1	<u>L5</u>
<u>L4</u>	L1 and container\$ and (directed near graph)	18	<u>L4</u>
<u>L3</u>	L2 and (directed near graph)	1	<u>L3</u>
<u>L2</u>	L1 and (plurality near container\$)	35	<u>L2</u>
<u>L1</u>	707/\$.ccls.	14009	<u>L1</u>

END OF SEARCH HISTORY

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

End of Result Set



Generate Collection

Print

L3: Entry 1 of 1

File: USPT

Jan 28, 2003

DOCUMENT-IDENTIFIER: US 6513040 B1

TITLE: Method and apparatus for a model data structure for accessing and manipulating java beans

Detailed Description Text (9):

The flexibility of the data structure in the repository model permits several types of association between the ManagedBean Elements and the Managed Beans. For instance, there can be a tree structure organization, in which the Managed Bean Elements of one Managed Bean are distinct from the Managed Bean Elements of another Managed Bean. Or, there can be a directed graph, in which there are distinct Managed Beans, but in which certain Managed Bean Elements belong to more than one Managed Bean. It is also the case that a set of Managed Bean Elements can be common to all Managed Beans in a repository.

Current US Original Classification (1):707/10Current US Cross Reference Classification (1):707/103R

CLAIMS:

1. A method, implemented in a computer system, for creating a storage container for a persistent collection of items, comprising the steps of: storing an object-oriented object in said storage container accessible to a user in said computer system, said storage container having a plurality of related objects in a persistent state; and creating a collection of persistent files as a subclass of said object within said storage container wherein said persistent files represent different types of code and data for said object.
6. An apparatus for creating a storage container for a persistent collection of items, comprising: means for storing an object-oriented object in said storage container accessible to a user in said computer system, said storage container having a plurality of related objects in a persistent state; and means for creating a collection of persistent files as a subclass of said object within said storage container wherein said persistent files represent different types of code and data for said object.
11. A computer program product having a computer readable medium having computer program logic recorded thereon for creating a storage container for a persistent collection of items, comprising: computer readable means for storing an object-oriented object in said storage container accessible to a user in said computer system, said storage container having a plurality of related objects in a persistent state; and computer readable means for creating a collection of persistent files as a subclass of said object within said storage container wherein said persistent files represent different types of code and data for said object.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

interest taxonomy tags is a measure of a knowledge container's relevance to the user's interests. Upon completion of the ranking step, a ranked list of knowledge containers is presented to the user. This completes an instance of retrieving an appropriate answer from a corporate knowledge base of populated taxonomies in response to a query.

Detailed Description Text (136):

Thus far, this specification has described the algorithm for retrieving appropriate knowledge containers as a single query-response sequence. In other words, users type a question, perhaps augmented by initial taxonomy tags, interest taxonomy tags, and/or taxonomic restrictions (filters), and a single list of knowledge containers is returned. Another aspect of the invention is the ability to use the taxonomies and the retrieval algorithm to create a multi-step interactive "dialog" with users that leads them to appropriate knowledge containers.

Detailed Description Text (139):

b) desired administrative meta-data values; e.g. desired date ranges for creation-date of knowledge containers to be retrieved,

Detailed Description Text (141):

d) taxonomic restrictions, used as described above (with respect to retrieval techniques) to limit the areas of taxonomies from which response knowledge containers are drawn.

Detailed Description Text (143):

a) a list of result knowledge containers that are possible "answers" to the question, each with a relevance score between 0 and 1;

Detailed Description Text (144):

b) a structured list of taxonomies, taxonomy regions, and/or taxonomy tags that the system believes may be associated with the question, and the weight of the association. This list may be augmented with annotations that indicate concept nodes, regions, or taxonomies that are likely to be mutually exclusive, e.g. because their knowledge containers use different vocabulary; and

Detailed Description Text (147):

(1) Knowledge container results: a list of zero or more knowledge containers that the system considers possible "answers" or highly relevant information to the user's question. These can be presented as clickable links with meta-data indicating the knowledge container's title, synopsis, dates, author, etc., where clicking will lead the user to a screen presenting the full content of the knowledge container; alternatively, if the system has one or more knowledge containers that it believes with high confidence will serve as answers to the user's question, it can simply display the full content of those knowledge containers directly.

Detailed Description Text (149):

All of these clarifying dialog techniques make significant and direct use of the multi-taxonomy structure that knowledge containers have been tagged into. The novel aspect exists in the combination of using a multi-taxonomy structure to tag knowledge containers via autocontextualization; to retrieve knowledge containers using the retrieval methods described above; and to drive an interactive dialog to help users find knowledge containers through multiple steps.

Detailed Description Text (150):

The combination of taxonomies, taxonomy tags, taxonomic restrictions (filters), and knowledge containers provide unequalled personalization capabilities to the present system. Certain of these taxonomies can be used to: capture the universe of information needs and interests of end-users; tag the knowledge containers representing these users with the appropriate concept nodes from these taxonomies, and use these concept nodes when retrieving information to personalize the delivery of knowledge containers to the user. Further, the system can use this tagging and other aspects of the knowledge containers in order to create a display format appropriate for the needs of the user receiving the knowledge container.

Detailed Description Text (151):

In order to personalize interactions with a specific customer, the system has a model for representing that customer and their interests and needs. As discussed above, that model is the knowledge container of type "Customer." The taxonomy tags associated with each customer

knowledge container specify what the customer is interested in, and how interested he or she is. The system supports profiling a customer's interaction with the system explicitly based on stated or applied preferences, and implicitly based on what the system has learned from interacting with the customer.

Detailed Description Text (152):

Explicit profiling allows the user to select items of interest explicitly from one or more taxonomies. These, along with a default or explicit weight, become taxonomy tags for their customer knowledge container. Implicit profiling, on the other hand, relies on the system to add or modify customer knowledge container taxonomy tags in order to profile the customer. For example, when creating the customer knowledge container, the system may set a concept in "access level" or "entitlement level" taxonomies that match the privileges they wish to accord the end user whom the knowledge container represents. The system may alternatively observe user behavior and then modify taxonomy tags accordingly. That is, the system can increase the weight of taxonomy tags that are frequently spotted in the user's questions during the autocontextualization segment of the retrieval process and it can increase the weight of taxonomy tags for answers given by the user during the dialog segment of the retrieval process. Finally, the business context of the interaction, including the application screen, can create an implicit profiling which drives the retrieval. For example, a particular web page or email address from which or to which a question is entered into the system may implicitly add taxonomy tags to the user's question. This particular kind of implicit profiling is typically transient in that it only modifies the current interaction, but does not change the tagging of the user's customer knowledge container.

Detailed Description Paragraph Table (1):

TABLE 1 Knowledge Document Container Type Represents Some kind of electronic content, typically with a text component. Usage Represents documents, their content and their meta-data. Knowledge Question Container Type Represents A question asked by a system end-user Usage Used to hold a question, whether to be automatically answered or forwarded to an expert. Questions maintain links to their Answers. Knowledge Answer Container Type Represents An answer to a question Usage Used to hold an answer created by an expert. Answers maintain links to their Questions. Knowledge PAQ Container Type Represents A previously asked question (PAQ) Usage Used to hold a question and its answer pre-packaged for reuse. Can be automatically returned in answer to a Question. PAQs maintain links to a Question and an Answer. Knowledge Knowledge Consumer Container Type Represents A person who uses knowledge containers, by browsing the knowledge base, by asking questions, etc. Usage Used to represent a system end-user, generally a customer or partner of the enterprise, but also internal users (experts etc.). Knowledge Consumer knowledge container taxonomy tags represent the person's interest areas and levels of interest in those areas. Knowledge Knowledge Provider Container Type Represents A person who contributes knowledge to the system. Usage Used to represent subject experts in the organization who contribute knowledge containers, route or answer questions, review answers, and edit the collection of knowledge containers. Knowledge Provider knowledge container taxonomy tags represent the expert's areas of expertise and level of expertise in those areas. Since Knowledge Providers also use knowledge, they are linked to a Knowledge Consumer knowledge container. Knowledge E-Resource Container Type Represents Some kind of electronic resource Usage Holds a description of and a link to an electronic resource, such as an online community of interest, a transactional web page, an application, a search engine, or any other addressable resource (e.g. addressable by a Uniform Resource Locator (URL)). Knowledge Product Container Type Represents A specific product or product family sold by the enterprise employing the system Usage Holds a description of a product or product family. Tags and meta-data indicate features of the product. The content of the knowledge container may take the form of a "catalog" entry that describes the product in text, includes a picture or "banner ad" for the product, etc.

Detailed Description Paragraph Table (2):

Tag the knowledge container to IBM if the knowledge container-content contains "<Company>IBM</Company>" or "<Company>International Business Machines</Company>".

Detailed Description Paragraph Table (3):

TABLE 2 Knowledge Container Meaning of a tag to a Concept Node in a: Type Topic Taxonomy Filter Taxonomy Lexical Taxonomy Question- Question's content is Tags indicate meta-data Content includes KC about the topic about or entitlements of mentions of the represented by the the question that tagged concept-nodes concept-node; weight of corresponds to the the tag indicates

the concept-node strength of the topic Document Document's content is Tags indicate meta-data Content includes KC about the topic about the that mentions of the represented by the corresponds to the tagged concept-nodes concept-node; weight of concept-node or the tag indicates the entitlements required to strength of the topic retrieve or view it Consumer-Consumer is interested in Tags indicate meta-data Consumer is KC (e.g. the topic represented by about the consumer that interested in the topic customer or the concept-node; weight corresponds to the represented by the other user of of the tag indicates concept-node or concept-node; weight knowledge) strength of the interest entitlements held by the of the tag indicates consumer strength of the interest Provider- Provider has expertise in Tags indicate meta-data Provider has expertise KC (expert) the topic represented by about the expert that in the topic the concept-node; weight corresponds to the represented by the of the tag indicates level concept-node or concept-node; weight of expertise entitlements required to of the tag indicates escalate to the user level of expertise Not applicable

Current US Original Classification (1):

707/104.1

CLAIMS:

2. The knowledge map of claim 1, wherein said graph is a directed acyclic graph.

17. The knowledge map representation of claim 15, wherein: one or more knowledge containers are associated with at least some of the nodes; and the nodes that are a member of the region have a similarity of vocabulary in the content of their associated knowledge containers.

18. The knowledge map representation of claim 15, wherein: the knowledge map region is generally centered about a particular central node; one or more knowledge containers are associated with at least some of the nodes; and the nodes that are members of the region are (1) those nodes having less taxonomic distance from the particular central node than nodes that are not members of the region, and also (2) for which there is similar vocabulary in the content of their. associated knowledge containers.

25. The knowledge map representation of claim 1, wherein at least one of said taxonomies comprises a standard taxonomy, in which each node of the standard taxonomy represents one or more topics, and an association of a knowledge instance with a node of the standard taxonomy indicates that at least some content of the knowledge container concerns the topic represented by the node.

34. The knowledge map of claim 32, further comprising at least one association between a knowledge container and a node, and wherein the association classifies the knowledge container within the knowledge domain.

Previous Doc

Next Doc

Go to Doc#

Freeform Search

Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Term:

L10 and ("run-time")

 Display: Documents in Display Format: Starting with Number

 Generate: ☐ Hit List ☒ Hit Count ☐ Side by Side ☐ Image

Search

Clear

Interrupt

Search History

 DATE: Friday, December 10, 2004 [Printable Copy](#) [Create Case](#)

<u>Set Name</u> side by side	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u> result set
<i>DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR</i>			
<u>L11</u>	L10 and ("run-time")	8	<u>L11</u>
<u>L10</u>	L8 and (library)	15	<u>L10</u>
<u>L9</u>	L8 and (information near library)	0	<u>L9</u>
<u>L8</u>	L7 and action	23	<u>L8</u>
<u>L7</u>	L3 and type	32	<u>L7</u>
<u>L6</u>	L5 and graph	37	<u>L6</u>
<u>L5</u>	L2 and (container near type)	121	<u>L5</u>
<u>L4</u>	L3 and (container near type)	4	<u>L4</u>
<u>L3</u>	L2 and (directed near graph)	34	<u>L3</u>
<u>L2</u>	L1 and (container\$)	1724	<u>L2</u>
<u>L1</u>	707/\$.ccls.	23917	<u>L1</u>

END OF SEARCH HISTORY

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

End of Result Set



Generate Collection

Print

L16: Entry 6 of 6

File: USPT

Mar 6, 2001

DOCUMENT-IDENTIFIER: US 6199068 B1

**** See image for Certificate of Correction ****

TITLE: Mapping interface for a distributed server to translate between dissimilar file formats

Drawing Description Text (34):

FIG. 33 illustrates slot names within a blackboard object that contain the specific value types used to execute the operations; and

Detailed Description Text (61):

The Activity Plan is a decision tree of these operations along with contextual information carried for the flow and available to each operation. The Activity Plan also defines which operations are dependent upon others and thus, which operations can run in parallel. Services within the activity dispatcher instantiate (start) an Activity Plan, negotiate responses and events for Activity Plans, and monitor the current status of all Activity Plans in progress. Activity Plans themselves are scripted outside the coding environment and are easily modified to tailor the AMR Server 15 for a particular client's business requirements. Thus, the business requirements may be easily changed without re-coding the underlying services and objects. The decision process for guiding execution is controlled by a directed graph of business logic encapsulated in each Activity Plan. The Activity Plan object represents a state machine that is self-directed. The dispatcher simply provides the Activity Plan objects an environment in which to execute.

Detailed Description Text (62):

The tasks have the following responsibilities. The first is task sequencing, which determines which tasks can be run in parallel vs. serial. The second responsibility is blackboard management, which holds and manages access to the blackboard for all contained tasks. The third is task state management, which tracks which tasks are in progress. Another responsibility is a next operation which is a directed graph rule-set for determining which task to perform next based on the state of the Activity Plan. The activity plans are also responsible for task logging, which logs the result of tasks as they are completed.

Detailed Description Text (69):

The Activity Plan Builder 146d is provided because Activity Plans are not useful objects immediately after instantiation. They are constructed and passivated for later use because Activity Plans are the objects that manage a set of tasks to perform a unit of business work. In addition, the Activity Plan object itself is simply a manager and container for the tasks that get the work done. An ordered collection of tasks are constructed and assigned to the Activity Plan before it is useful.

Detailed Description Text (83):

FIG. 33 illustrates the slot names within the blackboard object that contain the specific value types used to execute the operations.

Detailed Description Text (224):

The Validation, Editing, Estimation (VEE) Manager 152a is responsible for performing the validation, editing, and estimation specified by a particular Regulatory Agency to produce settlement quality data for export from the AMR Server 15. As with all Encina.RTM. Servers in the system, the VEE Manager 152a uses the AppServer classes to receive service requests through RQS. The VEE Manager 152a uses a directed graph and the performer to execute different functions. Each request is for VEE 152a on a particular meter/rate combination and will be executed within its own thread. Although shown logically as existing within the Export Subsystem 152, the VEE Manager 152a is actually contained within the same process space as the

Reading Manager. The VEE Manager 152a will nonetheless provide a separate interface and be bound to as if it was a separate server. It physically resides with the Reading Manager as a performance optimization to minimize the transport of data across the network and benefit from local Persistence object caching. FIGS. 34A-D illustrate the various threads executing in the VEE 152a.

Detailed Description Text (228):

In order to accomplish validation, editing, and estimation the VEE Manager 152a will use local Activity Plans and a local dispatcher to run these plans. This Local Dispatching approach has been designed for use in VEE 152a to take advantage of the fact that all primary objects used in VEE 152a are in the same process space. The Local Dispatcher performs a Local Activity Plan which only executes Local Operations that carry out actions on local objects. Local operations generate a workflow slot, and a ForcedRereadNeeded, which indicates the need to reread the physical meter 60 or communication server 30 to retrieve more accurate readings for a specified time period and then reapply the readings to the VEE 152a. All parameters are in the blackboard. Other batched services may use the Local Dispatching approach for performance enhancement, if they also depend strictly on Local objects performing synchronously. This implementation uses a modified version of the infrastructure developed for the Activity Management Subsystem 146. The directed graph logic will contain the Regulatory Agency specific tasks and rules.

Current US Original Classification (1):

707/100

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Generate Collection

Print

L4: Entry 9 of 18

File: USPT

Oct 23, 2001

DOCUMENT-IDENTIFIER: US 6308181 B1

TITLE: Access control with delayed binding of object identifiers

Brief Summary Text (7):

In some hierarchical databases, objects may also be grouped in containers. A particular container object serves as the root of a particular subtree. One use of subtrees is to control access to objects in the database, as discussed below.

Brief Summary Text (13):

In some systems, rights granted to "object rights" or "all properties rights" may be inherited. For instance, rights granted at a container may also apply to all objects in the subtree of which the container is the root.

Drawing Description Text (15):

FIG. 13 is a diagram illustrating part of a directed-graph-structured database having objects at its vertices, with access to the objects being controlled through use of access control properties according to the present invention.

Detailed Description Text (4):

Suitable databases for adaptation according to the invention include those presently configured to use ACLs or compatible with ACLs, such as many directory service databases generally, Novell Directory Service databases in particular, and other databases (hierarchical or otherwise positionally structured) having components such as containers, leaves, objects, and/or attributes to which access is controlled by ACLs or other associations between a target object, a trustee object, and access rights.

Detailed Description Text (18):

A database need not be hierarchical to have positional relationships. For instance, FIGS. 12 and 13, which are discussed in detail later, illustrate positional relationships in a graph rather than a tree. In a graph, positional relationships may include relationships such as "immediate neighbor" or "neighbor at most N steps away". In a directed graph, positions may also reflect link directions, such as "reachable neighbor" or can be reached from in at most N hops".

Detailed Description Text (61):

Although the discussion up to this point has focused on the positional relationship evaluating to a value that makes sense within the context of a hierarchical database, such as the relationship "child", in alternative embodiments positional relationships can also be evaluated to values that make sense in other configurations, such directed graphs and undirected graphs.

Detailed Description Text (63):

Graphs in general, and directed graphs in particular, are well known in the computer science and programming arts. Some of the positional relations used with undirected graphs and/or directed graphs can be employed according to the present invention; a few of the many possible examples are shown in FIGS. 12 and 13.

Detailed Description Text (64):

As shown in these two Figures, each graph includes one or more "vertices" (also known as "nodes" or "points" or "locations") connected with zero or more "arcs" (sometimes "edges" or "links" or "segments"). A directed graph is a graph in which arcs may have a direction, as indicated by the arrowheads in FIG. 13. Vertices separated by exactly one edge are adjacent. For instance, vertex 1200 and vertex 1202 are adjacent, while vertex 1200 and vertex 1206 are not adjacent. In directed graphs, direction may be made part of the definition of adjacency, so

that vertex 1300 is adjacent to vertex 1304 but vertex 1304 is not adjacent to vertex 1300, for example. Those of skill in the art will recognize that other terminology is also used with graphs to discuss concepts such as connectivity, distance in hops, spanning, and so on.

Detailed Description Text (69):

A database 1406 is stored in the memory 1404. The database 1406 may be a directory services depository, a hierarchical database, a synchronized-partition database, an object-oriented-relational-hybrid database, a collection of objects arranged in a graph or a directed graph or a tree, or some combination of such databases.

Detailed Description Text (79):

The invention may be embodied in other specific forms without departing from its essential characteristics, including object databases which are only partially hierarchical, and object databases structured as undirected graphs or directed graphs. The described embodiments are to be considered in all respects only as illustrative and not restrictive. Any explanations provided herein of the scientific principles employed in the present invention are illustrative only. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

Current US Original Classification (1):

707/102

CLAIMS:

17. The method of claim 1, wherein the chosen positional relationship is a directed graph relationship comprising at least one relationship from the group of relationships containing "adjacent" and "reachable from" relationships.

25. The system of claim 24, wherein database comprises objects arranged in a directed graph structure.

[Previous Doc](#)

[Next Doc](#)

[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Generate Collection

Print

L4: Entry 14 of 18

File: USPT

Nov 25, 1997

DOCUMENT-IDENTIFIER: US 5692180 A

TITLE: Object-oriented cell directory database for a distributed computing environment

Abstract Text (1):

A memory for storing data for access by an application program being executed in a processing system of a distributed computing environment includes an object-oriented data structure. The data structure include a plurality of base data objects having predetermined instances including a directory object and an entry object, and a plurality of attribute data objects storing information about particular resources. The plurality of base and attribute data objects are configurable into a namespace, e.g., a tree structure, that enables the application program to locate resources in the distributed computing environment. The predetermined instances of the base data objects may also include a link object that points to another object in the tree structure, thereby transforming the namespace into a directed graph.

Brief Summary Text (15):

According to the invention, every resource in the cell has a unique name. These names are actually objects in memory, and the objects are organized into a namespace. Each object may include attributes that detail information about the resource. To facilitate the directory service, each object is structured into one of several types, a directory, a link or an entry, and objects are organized into a tree structure that is configurable into a directed graph. A directory object is a container to hold resource names, names of other directories or names of other object types. A link object is used to point to another name, so that a reference to the link actually returns references to the link's target. An entry object is a leaf object in a naming tree.

Brief Summary Text (16):

Preferably, the common qualities of the three types of objects are abstracted into a common base object. Each base object contains a list of children, and a parent pointer. These elements enable the base objects to be organized into the tree structure. By adding link objects, the tree structure may be transformed into a general directed graph. Children of a base object are strung together in a singly linked-list structure.

Detailed Description Text (11):

As described generally above, the database of names and other server information is stored in a namespace defining a tree structure that is configurable into a directed graph. The database provides DCE distributed applications with a place to store and find information about resources available in the cell. One representative use of the database is to provide DCE distributed application clients with a remote server's network address. Besides server location information, the naming service may provide other information including, without limitation, the server's name, the protocols supported by a server, the types of services available, and other interface information. This information is exported to the directory server whenever an application server starts up.

Detailed Description Text (12):

Every resource (typically an application server) in the cell has a unique name. According to the invention, the names are preferably objects in memory formatted under C++ programming language conventions. Although not meant to be limiting, the namespace provides for the following types of objects: CDS directory, CDS link and CDS entry. A CDS directory is a container to hold additional names, links or entries, or references to subdirectories. A CDS directory contains zero or more children. A link object is used to point to another name so that references to the link returns a reference to the link's target. An entry object is a leaf object in a naming tree.

Detailed Description Text (14):

Therefore, CDS objects may be directories (containing other CDS objects), a link (a pointer to another CDS object), or an entry (a leaf object). A CDS object may contain zero or more attributes, and every attribute belongs to a CDS object. The common qualities of the three types of CDS objects (the directory, link and entry objects) are abstracted into a common CDS base object, described here by a C++ object called cdsBase. Each cdsBase object contains a list of children and a parent pointer. Such elements enable the cdsBase objects to be organized into a tree structure. In addition, adding link support (through the link object) allows the tree structure to transform into a more general directed graph, which may be cyclic. The children of a cdsBase object are strung together in a singly linked-list structure. As discussed above, each cdsBase object may also contain attributes, and each attribute is modeled by a C++ object called cdsAttr. The cdsAttr objects are strung together in a singly linked-list.

Detailed Description Text (15):

FIG. 4 shows a representative namespace with a root directory "/. . ." and a directory "mycell". The namespace also includes a CDS entry named "/. . ./mycell/myobj" and a CDS directory named "/. . ./mycell/mydir." As can be seen, there is one attribute on entry "/. . ./mycell/myobj". The name of this attribute is "CDS.sub.-- Class" and its value is the string "RPC.sub.-- Entry." With the addition of link objects, the tree structure is transformable to a directed graph. A link object includes a pointer that provides an alternate for an object entry, directory or other link in the directory service database. CDS link objects restructure the directory hierarchy by pointing from an existing name to a new name. This is also illustrated in FIG. 4. In particular, the name "/. . ./mycell/mylink" is considered another name for 35 "/. . ./mycell/myobj", and there is one attribute on this object named "CDS.sub.-- LinkTarget". When accessing a soft link, CDS will automatically read the CDS link target attribute and access information in the target object.

Current US Original Classification (1):

707/10

Current US Cross Reference Classification (1):

707/100

CLAIMS:

3. The memory as described in claim 1 wherein the namespace is a directed graph.

12. In the network as described in claim 10 wherein the namespace of the data structure in the memory is a directed graph.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)